Exercise 1 (6 Points)

In the following we consider a grammar $G$ whose derivation tree visualizes a Christmas tree.

Let $G = (N, \Sigma, P, R)$ be the CFG with $N = \{R, T, B\}$ and $\Sigma = \{\text{star, candle, red, blue}\}$.

The productions $P$ are as follows:

$$
R \rightarrow T \\
T \rightarrow BTB \mid T \astar \\
B \rightarrow BBB \mid BB \mid B \mid \text{blue} \\
$$

An example Christmas tree (upside down with the root at the top) would look as follows:

![Christmas Tree Diagram]

Give an attribute grammar $A = (G, E, V)$ for $G$ such that the obtained Christmas trees (i.e., derivation trees) satisfy the following restrictions:

(i) The distance from $\astar$ to the start symbol $R$ is greater than the distance between every other leaf and $R$.

(ii) The number of $\astar$ on the left and right hand side of the tree (split by the trunk $T$) are equal.

(iii) When traversing the tree in a pre-order manner the $\astar$ and $\text{blue}$ are encountered alternatingly and the first visited ornament is $\astar$.

*Hint:* Pre-order is a depth-first traversal where first the node, then the left subtree and last the right subtree are visited.

As before, you may use a synthesized Boolean attribute $b$ such that: A word $w$ is in the language of $A = (G, E, V)$ if $w \in L(G)$ and, at the root of a derivation tree, $b = true$.

The given example Christmas tree satisfies all three restrictions as can be easily seen:

(i) The distance from $\astar$ to the start symbol $R$ is 6 whereas the largest other distance is 5.

(ii) On the left hand side there is one $\text{blue}$ and on the right hand side there is one $\text{blue}$ as well.

(iii) The traversal yields the following (correct) order of ornaments: $\astar, \astar, \astar, \astar$.

---

1For ease of notation you can also use $\Sigma = \{\text{star, candle, red, blue}\}$. 

---
Exercise 2

(4 Points)

Consider the following intermediate code:

```
2: LIT(-1);  
3: LOAD(1, 3); (dif, off)  
4: LT;  
5: JFALSE(7);  
6: CALL(6, 1, 3); (ca, dif, loc)  
7: RET;
```

Give the next four states of the abstract machine starting in:

```
```

Recall that the procedure stack has the form:

```
st | dl | ra | v1 | . . . | vn | . . .
```

and the base-function is defined as:

```
base(p, 0) := 1
base(p, dif + 1) := base(p, dif) + p.base(p, dif)
```

Exercise 3

(4 Points)

In addition to `while`-loops we want to have `for`-loops with implicit declaration of the counter variable in our example programming language:

```
for (var X := A ; B ; C1 ) C2
```

(a) Extend the translation function `ct` accordingly. You may assume that the variable `X` is already declared, i.e., it is `update(var X, st, l)` with `st` the symbol table and `l` the current level.

(b) Generate intermediate code for

```
for (var x := 0; x < 10; x := x + 1) P()
```

without parameters for the `CALL` instruction generated for `P()`.

We wish you a merry Christmas and
a Happy New Year!